

WHAT IS CLAIMED IS:

1. A system, comprising:
a storage device configured to store a plurality of files; and
5 a file system configured to manage access to said storage device and to store file system content, wherein said file system comprises a programming-language-independent interface whereby an application accesses said file system content.
- 10 2. The system as recited in claim 1, wherein said file system content comprises file data stored in one or more of said plurality of files.
3. The system as recited in claim 1, wherein said file system content comprises metadata stored in a named stream corresponding to a given file.
- 15 4. The system as recited in claim 1, wherein said metadata is stored in Extensible Markup Language (XML) format.
5. The system as recited in claim 1, wherein said programming-language-independent interface is configured to:
20 detect a virtual file identity corresponding to a given file;
select at least a portion of said file system content dependent on said virtual file identity; and
return said selected file system content.
- 25 6. The system as recited in claim 5, wherein said virtual file identity is formed by embedding a command token within a file identity corresponding to said given file.

7. The system as recited in claim 5, wherein said virtual file identity is formed by prepending a virtual directory to a file identity corresponding to said given file.

8. A method, comprising:
5 storing file system content; and
accessing said file system content via a programming-language-independent interface.

9. The method as recited in claim 8, wherein said file system content comprises file
10 data stored in one or more files.

10. The method as recited in claim 8, wherein said file system content comprises metadata stored in a named stream corresponding to a given file.

15 11. The method as recited in claim 8, wherein said metadata is stored in Extensible Markup Language (XML) format.

12. The method as recited in claim 8, wherein said programming-language-independent interface is configured to:
20 detect a virtual file identity corresponding to a given file;
select at least a portion of said file system content dependent on said virtual file identity; and
return said selected file system content.

25 13. The method as recited in claim 12, wherein said virtual file identity is formed by embedding a command token within a file identity corresponding to said given file.

14. The method as recited in claim 12, wherein said virtual file identity is formed by prepending a virtual directory to a file identity corresponding to said given file.

15. A computer-accessible medium comprising program instructions, wherein the
5 program instructions are computer-executable to:
store file system content; and
access said file system content via a programming-language-independent
interface.

10 16. The computer-accessible medium as recited in claim 15, wherein said file system content comprises file data stored in one or more files.

17. The computer-accessible medium as recited in claim 15, wherein said file system content comprises metadata stored in a named stream corresponding to a given file.

15

18. The computer-accessible medium as recited in claim 15, wherein said metadata is stored in Extensible Markup Language (XML) format.

19. The computer-accessible medium as recited in claim 15, wherein said
20 programming-language-independent interface is configured to:
detect a virtual file identity corresponding to a given file;
select at least a portion of said file system content dependent on said virtual file
identity; and
return said selected file system content.

25

20. The computer-accessible medium as recited in claim 19, wherein said virtual file identity is formed by embedding a command token within a file identity corresponding to said given file.

21. The computer-accessible medium as recited in claim 19, wherein said virtual file identity is formed by prepending a virtual directory to a file identity corresponding to said given file.

5